Lab Report 3

Sonnelly Cheong, Zhian Lin

```r
## Read data here
spotify <- read.csv("spotify_songs.csv", header=TRUE)

## Call Required Libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------ tidyverse_conflic
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library('qqtest')
library(gridExtra)
```

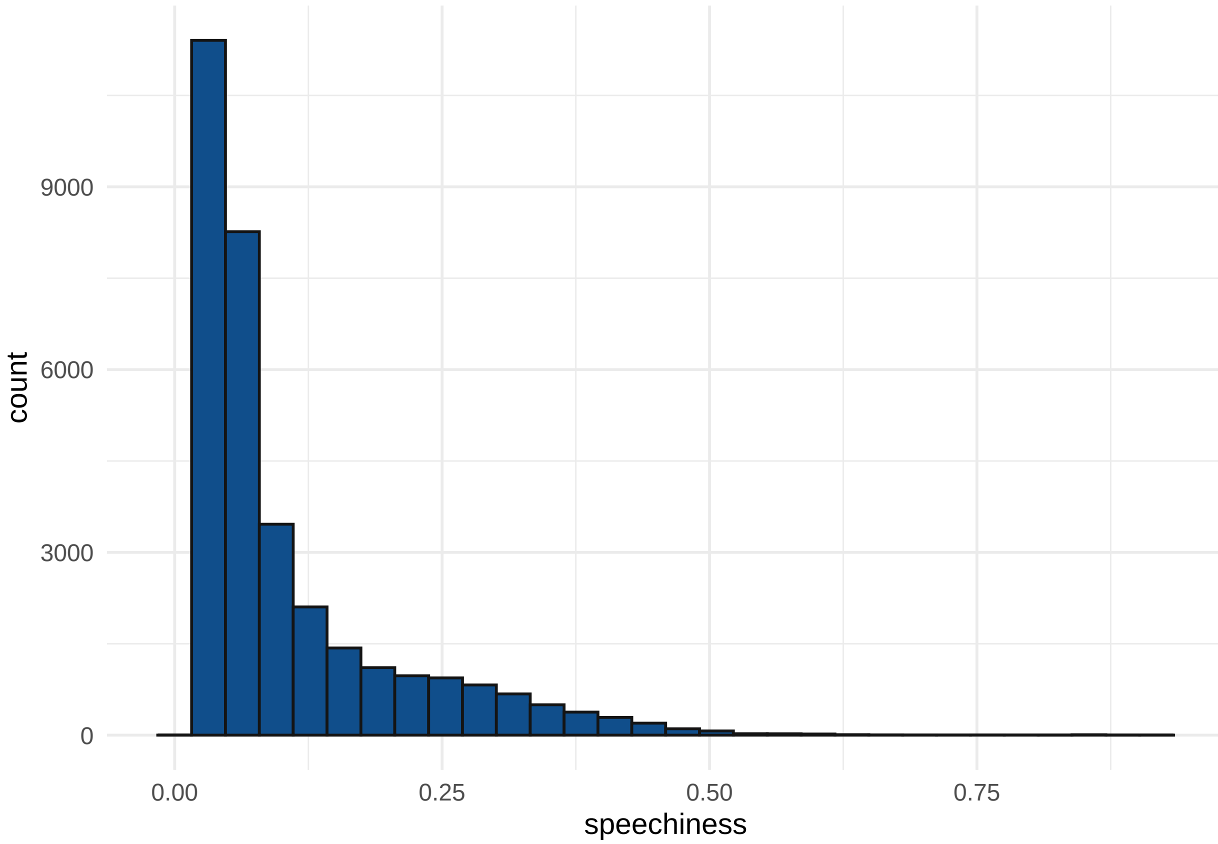
```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library('ggExtra')

## Set theme to something else (optional)
theme_set(theme_minimal())
```

1.Central Limit Theorem

```r
g1 <- ggplot(spotify, aes(x=speechiness)) +
  geom_histogram(fill = 'dodgerblue4', colour='gray8')

grid.arrange(g1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



> It is apparent that this continuous variable, speechiness is right skewed and therefore it is not in normal condition.

```r
set.seed(156)

Xbars <- sapply(1:300, function(i) mean(spotify$speechiness[sample(1:nrow(spotify), 200)]))
Xbars <- data.frame(1:length(Xbars), Xbars)
colnames(Xbars) <- c('GroupID', 'Xbar')
Xbars
```

```
##     GroupID      Xbar
## 1         1 0.1043500
## 2         2 0.1056795
## 3         3 0.1077285
## 4         4 0.0992095
## 5         5 0.1050190
## 6         6 0.1077465
## 7         7 0.1126770
## 8         8 0.1021435
## 9         9 0.1179630
## 10       10 0.1093420
## 11       11 0.1235460
## 12       12 0.1001910
## 13       13 0.1078685
## 14       14 0.1077770
## 15       15 0.1040295
## 16       16 0.0925210
## 17       17 0.1001875
```

```
## 18          18 0.1020045
## 19          19 0.1058290
## 20          20 0.1051645
## 21          21 0.1119250
## 22          22 0.0945400
## 23          23 0.1080210
## 24          24 0.1025115
## 25          25 0.0977535
## 26          26 0.1071355
## 27          27 0.1075370
## 28          28 0.1091345
## 29          29 0.1064010
## 30          30 0.1026490
## 31          31 0.1146930
## 32          32 0.1159175
## 33          33 0.1012605
## 34          34 0.1114875
## 35          35 0.0954830
## 36          36 0.1078585
## 37          37 0.1051840
## 38          38 0.1058665
## 39          39 0.1088465
## 40          40 0.1141235
## 41          41 0.1100410
## 42          42 0.1105540
## 43          43 0.1179070
## 44          44 0.1013025
## 45          45 0.0985145
## 46          46 0.1079960
## 47          47 0.1145315
## 48          48 0.1089985
## 49          49 0.1048930
## 50          50 0.0991255
## 51          51 0.1061640
## 52          52 0.0991415
## 53          53 0.1145335
## 54          54 0.1018295
## 55          55 0.1197315
## 56          56 0.1093925
## 57          57 0.0996040
## 58          58 0.0975705
## 59          59 0.1084175
## 60          60 0.1193815
## 61          61 0.1152335
## 62          62 0.1189720
## 63          63 0.1006575
## 64          64 0.1071870
## 65          65 0.1087210
## 66          66 0.1217825
## 67          67 0.0928280
```

```
## 68         68 0.1114330
## 69         69 0.1096470
## 70         70 0.1047525
## 71         71 0.1031885
## 72         72 0.0941510
## 73         73 0.1238570
## 74         74 0.1031880
## 75         75 0.1095565
## 76         76 0.1048860
## 77         77 0.1134365
## 78         78 0.0969770
## 79         79 0.1023160
## 80         80 0.1076095
## 81         81 0.0993100
## 82         82 0.0906455
## 83         83 0.1037075
## 84         84 0.0969595
## 85         85 0.1163260
## 86         86 0.0974825
## 87         87 0.1104805
## 88         88 0.1149495
## 89         89 0.1126075
## 90         90 0.1069540
## 91         91 0.1065975
## 92         92 0.1096025
## 93         93 0.0974440
## 94         94 0.1115505
## 95         95 0.0852645
## 96         96 0.1137370
## 97         97 0.1090245
## 98         98 0.0984195
## 99         99 0.1216900
## 100       100 0.1082675
## 101       101 0.1124750
## 102       102 0.1058185
## 103       103 0.0947965
## 104       104 0.1098955
## 105       105 0.1025910
## 106       106 0.0968105
## 107       107 0.1086785
## 108       108 0.0895795
## 109       109 0.0905570
## 110       110 0.1276665
## 111       111 0.1107095
## 112       112 0.1016480
## 113       113 0.1159920
## 114       114 0.1081335
## 115       115 0.1144995
## 116       116 0.1092395
## 117       117 0.0928890
```

```
## 118      118 0.0963460
## 119      119 0.1090960
## 120      120 0.1237350
## 121      121 0.1066520
## 122      122 0.1062885
## 123      123 0.0999980
## 124      124 0.1092010
## 125      125 0.1116685
## 126      126 0.1136470
## 127      127 0.1026860
## 128      128 0.1068675
## 129      129 0.1037760
## 130      130 0.1100400
## 131      131 0.1027160
## 132      132 0.1085850
## 133      133 0.1115065
## 134      134 0.1106140
## 135      135 0.1012905
## 136      136 0.1057840
## 137      137 0.1040355
## 138      138 0.1137635
## 139      139 0.1065955
## 140      140 0.1022000
## 141      141 0.1131115
## 142      142 0.1015185
## 143      143 0.0961235
## 144      144 0.1032190
## 145      145 0.0978925
## 146      146 0.1066245
## 147      147 0.1087835
## 148      148 0.1008550
## 149      149 0.0949690
## 150      150 0.1041305
## 151      151 0.1167585
## 152      152 0.1060260
## 153      153 0.1035535
## 154      154 0.1172320
## 155      155 0.1035145
## 156      156 0.1100950
## 157      157 0.1033215
## 158      158 0.1025790
## 159      159 0.1087765
## 160      160 0.1164045
## 161      161 0.1076695
## 162      162 0.1120145
## 163      163 0.1018350
## 164      164 0.1127640
## 165      165 0.0982540
## 166      166 0.1094635
## 167      167 0.1176795
```

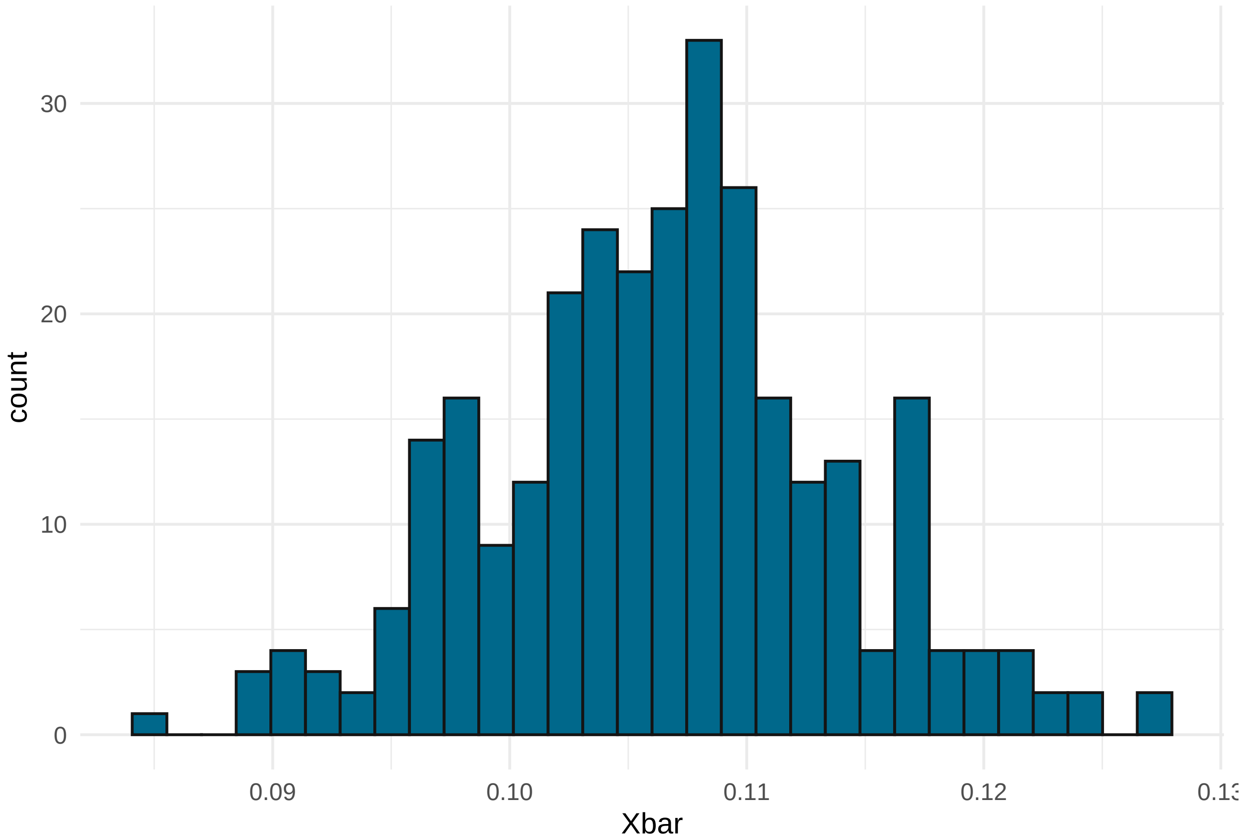
```
## 168      168 0.0959285
## 169      169 0.1116450
## 170      170 0.1050500
## 171      171 0.1020360
## 172      172 0.1221080
## 173      173 0.0960760
## 174      174 0.1071570
## 175      175 0.1045095
## 176      176 0.1069180
## 177      177 0.1183245
## 178      178 0.0961020
## 179      179 0.1109180
## 180      180 0.1088995
## 181      181 0.1035085
## 182      182 0.1008595
## 183      183 0.0962590
## 184      184 0.1083100
## 185      185 0.1082960
## 186      186 0.1060155
## 187      187 0.1176190
## 188      188 0.1045335
## 189      189 0.1104695
## 190      190 0.1034980
## 191      191 0.1175695
## 192      192 0.1173620
## 193      193 0.0980095
## 194      194 0.1164320
## 195      195 0.1200400
## 196      196 0.1003025
## 197      197 0.0968700
## 198      198 0.1033705
## 199      199 0.1051665
## 200      200 0.1029310
## 201      201 0.1049745
## 202      202 0.1092785
## 203      203 0.1069115
## 204      204 0.1110315
## 205      205 0.1125425
## 206      206 0.1106710
## 207      207 0.1140515
## 208      208 0.1013175
## 209      209 0.1131110
## 210      210 0.1145130
## 211      211 0.1009515
## 212      212 0.0984150
## 213      213 0.1207650
## 214      214 0.0968125
## 215      215 0.0958655
## 216      216 0.1064105
## 217      217 0.1054100
```

```
## 218       218 0.1019715
## 219       219 0.1176750
## 220       220 0.1170265
## 221       221 0.1043990
## 222       222 0.1099360
## 223       223 0.1016750
## 224       224 0.1072810
## 225       225 0.0983235
## 226       226 0.1125470
## 227       227 0.1102735
## 228       228 0.1103725
## 229       229 0.1061745
## 230       230 0.1123990
## 231       231 0.0965575
## 232       232 0.1061870
## 233       233 0.1200320
## 234       234 0.0947020
## 235       235 0.1064425
## 236       236 0.0992670
## 237       237 0.1060155
## 238       238 0.1163450
## 239       239 0.1051450
## 240       240 0.0919055
## 241       241 0.1102590
## 242       242 0.1075695
## 243       243 0.0911220
## 244       244 0.0978825
## 245       245 0.1110970
## 246       246 0.1084930
## 247       247 0.0986110
## 248       248 0.1079450
## 249       249 0.1173940
## 250       250 0.1176505
## 251       251 0.1049935
## 252       252 0.1088115
## 253       253 0.1038040
## 254       254 0.0987765
## 255       255 0.1081785
## 256       256 0.1164105
## 257       257 0.1083775
## 258       258 0.1020630
## 259       259 0.1038610
## 260       260 0.0907290
## 261       261 0.1068840
## 262       262 0.0986515
## 263       263 0.1052000
## 264       264 0.0962125
## 265       265 0.1106250
## 266       266 0.1125215
## 267       267 0.1076615
```

```
## 268     268 0.1100400
## 269     269 0.0973810
## 270     270 0.1052045
## 271     271 0.1018900
## 272     272 0.1100215
## 273     273 0.1275665
## 274     274 0.1043245
## 275     275 0.0999675
## 276     276 0.1094670
## 277     277 0.0884825
## 278     278 0.1058220
## 279     279 0.1064485
## 280     280 0.1041110
## 281     281 0.1140680
## 282     282 0.1050755
## 283     283 0.0897475
## 284     284 0.1027935
## 285     285 0.1101675
## 286     286 0.1081945
## 287     287 0.1049580
## 288     288 0.1072465
## 289     289 0.1146330
## 290     290 0.1037365
## 291     291 0.1022890
## 292     292 0.1079970
## 293     293 0.0982240
## 294     294 0.0950420
## 295     295 0.1090560
## 296     296 0.1039420
## 297     297 0.1076565
## 298     298 0.1211150
## 299     299 0.1075040
## 300     300 0.1168065
```

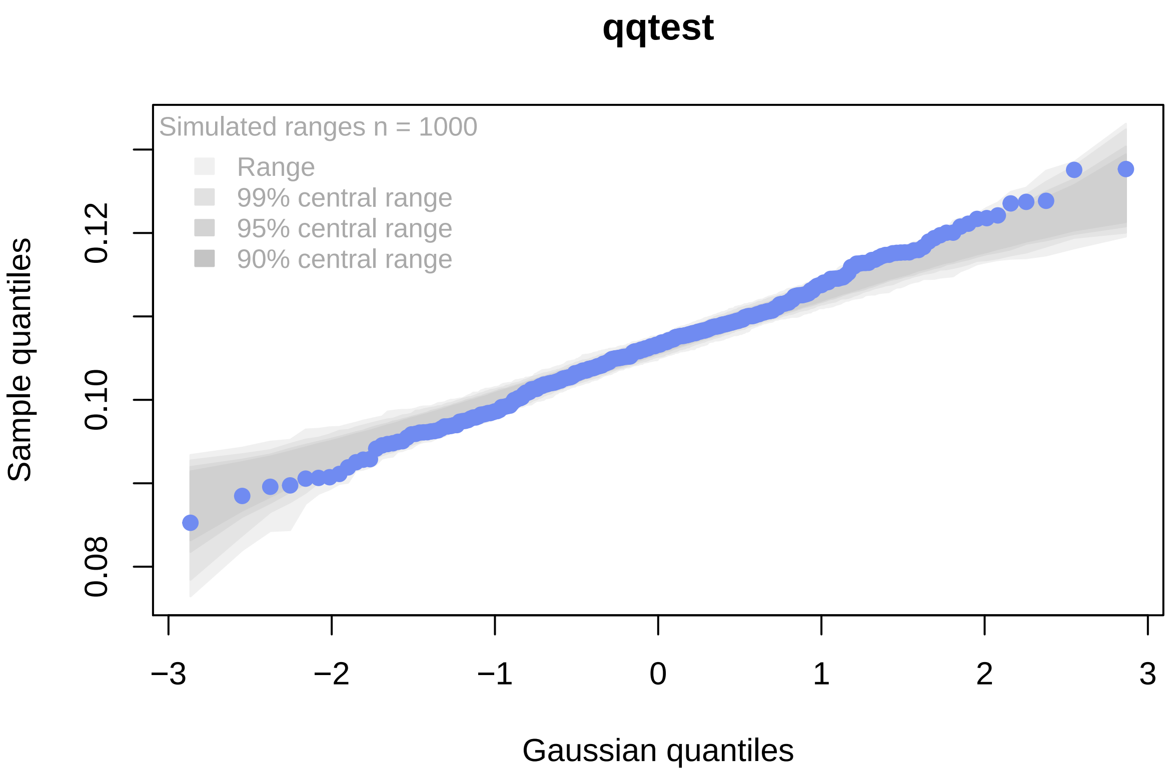
```r
ggplot(Xbars, aes(x=Xbar)) +
  geom_histogram(fill='deepskyblue4',colour='gray8')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



> The sample is now in normal distribution

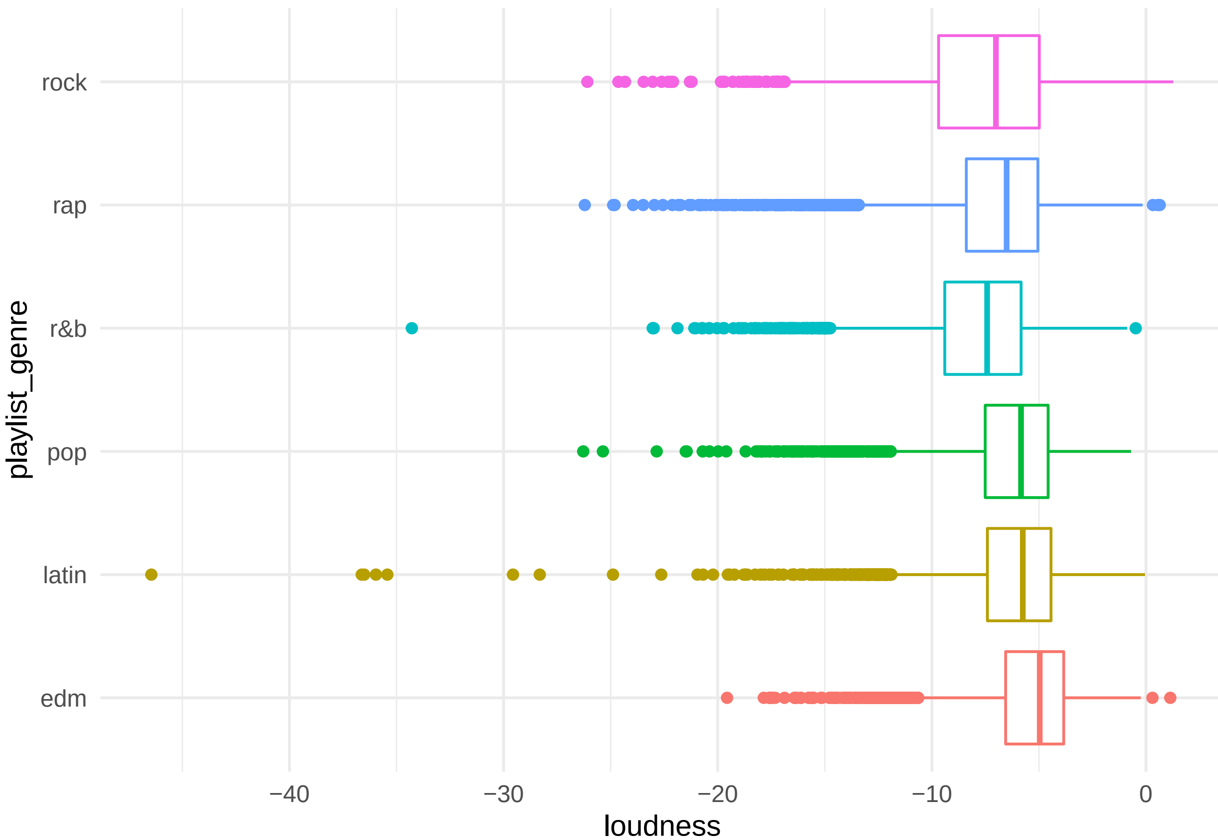
```
qqtest(Xbars$Xbar)
```



The test result indicates that the it fits within the envelope therefor it is safe to conclude that it is now a normal distribution.

2. Comparing Two Variable

```r
ggplot(spotify, aes(x =loudness, y=playlist_genre, colour = playlist_genre)) +
  geom_boxplot() +
  theme(legend.position = 'none')
```



Boxplot of the continuous variable (energy) on the x-axis and the categorical variable (energy) on the y-axis

```r
sample1 <- subset(spotify, playlist_genre == 'pop')$loudness
sample2 <- subset(spotify, playlist_genre == 'rap')$loudness
```

Hypothesis testing The first hypothesis: $H_0 : S_1^2 = S_2^2, S_1^2 - S_2^2 = 0$ tested as below:

```r
var.test(sample1, sample2)
```

```
##
##  F test to compare two variances
##
## data:  sample1 and sample2
## F = 0.737, num df = 5506, denom df = 5745, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6994663 0.7765636
## sample estimates:
## ratio of variances
##          0.7369967
```

Reject the hypothesis that the variance of loudness in pop and rap are equal

The second hypothesis: $H_0 : S_1^2 \geq S_2^2$ tested as below:

```r
var.test(sample1, sample2, alternative = 'less')
```

```
##
##  F test to compare two variances
##
## data:  sample1 and sample2
## F = 0.737, num df = 5506, denom df = 5745, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is less than 1
## 95 percent confidence interval:
##  0.0000000 0.7700609
## sample estimates:
## ratio of variances
##          0.7369967
```

Reject the hypothesis that the variance of loudness in pop is greater than rap

The third hypothesis: $H_0 : S_1^2 \leq S_2^2$ tested as below:

```r
var.test(sample1, sample2, alternative = 'greater')
```

```
##
##  F test to compare two variances
##
## data:  sample1 and sample2
## F = 0.737, num df = 5506, denom df = 5745, p-value = 1
## alternative hypothesis: true ratio of variances is greater than 1
## 95 percent confidence interval:
##  0.7053688       Inf
## sample estimates:
## ratio of variances
##          0.7369967
```

Cannot reject the hypothesis that the variance of loudness in pop is less than rap

3. Comparing two population means

The first hypothesis: $H_0 : \mu_1 = \mu_2 \Rightarrow \mu_1 - \mu_2 = 0$ tested as below

```
t.test(sample1,sample2, var.equal = F)
```

```
##
##  Welch Two Sample t-test
##
## data:  sample1 and sample2
## t = 13.564, df = 11118, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.6218882 0.8319943
## sample estimates:
## mean of x mean of y
## -6.315328 -7.042269
```

Reject the hypothesis that the mean loudness in two playlist genres are equal.

The second hypothesis: $H_0 : \mu_1 - \mu_2 \geq 0$ tested as below:

```
t.test(sample1,sample2, var.equal = F, alternative = 'less')
```

```
##
##  Welch Two Sample t-test
##
## data:  sample1 and sample2
## t = 13.564, df = 11118, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.8151023
## sample estimates:
## mean of x mean of y
## -6.315328 -7.042269
```

Cannot reject the hypothesis that the mean loudness of pop is larger than the mean of rap

The third hypothesis: $H_0 : \mu_1 - \mu_2 \leq 0$ tested as below

```
t.test(sample1,sample2, var.equal = F, alternative = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  sample1 and sample2
## t = 13.564, df = 11118, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.6387802       Inf
## sample estimates:
## mean of x mean of y
## -6.315328 -7.042269
```

Reject the hypothesis that the mean loudness of pop is less than the mean loudness of rap

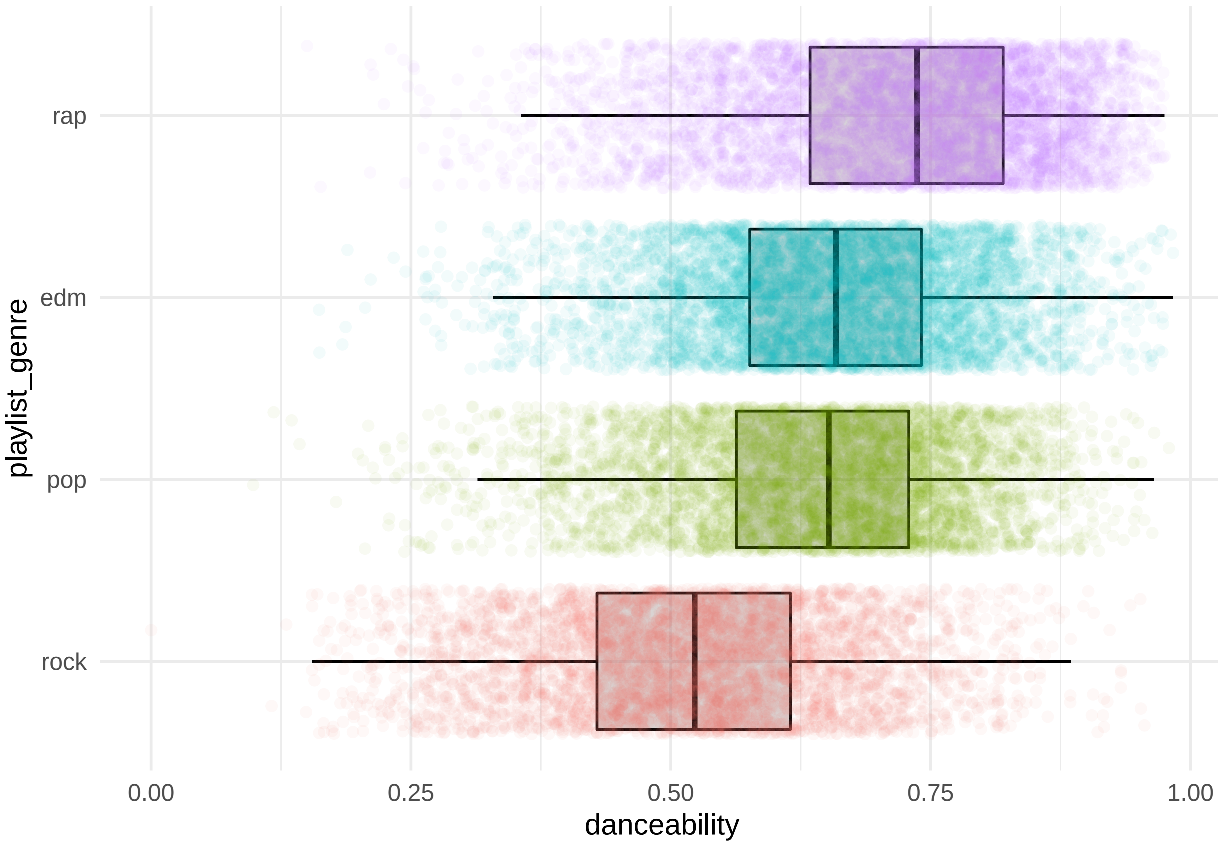
Question 4

We are interested in the danceability in 4 genres and trying to test whether the population means are equal. If the ratio between total variability within groups is not statistically different from variability among groups, then we cannot reject the hypothesis that the means are equal.

```
temp <- subset(spotify, playlist_genre %in% c('pop','rap','edm','rock'))
medians <- temp %>% group_by(playlist_genre) %>% summarise(medians=median(danceability))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
temp$playlist_genre <- factor(temp$playlist_genre,
  levels=medians$playlist_genre[order(medians$medians)])

ggplot(temp, aes(x=danceability,y=playlist_genre, colour=playlist_genre)) +
  geom_boxplot(outlier.alpha = 0, fill=adjustcolor('grey50',.3), colour='black') +
  geom_jitter(alpha=.05) +
  theme(legend.position = 'none')
```




```
anova(lm(danceability ~ playlist_genre,data = temp))
```

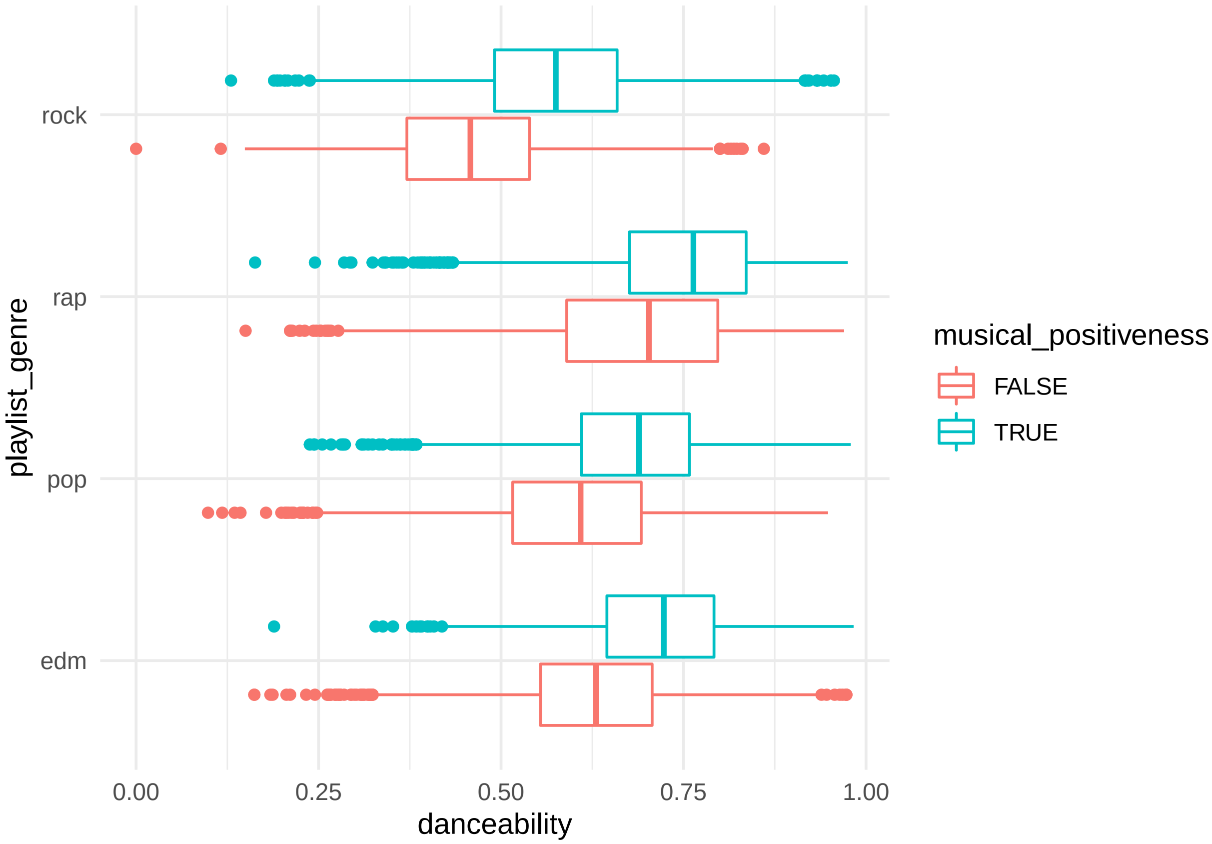
```
## Analysis of Variance Table
##
## Response: danceability
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## playlist_genre     3 107.16  35.719  2055.3 < 2.2e-16 ***
## Residuals      22243 386.56   0.017
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is too small, so we reject the hypothesis that the four playlist genres have equal population means.

Question 5

```r
temp <- subset(spotify, playlist_genre %in% c('pop','rap','edm','rock'))
temp$musical_positiveness <- temp$valence > 0.5

ggplot(temp, aes(x=danceability, y=playlist_genre, colour=musical_positiveness)) +
  geom_boxplot()
```



The valence describes the musical positiveness of a track. A higher valence value means that a song conveys more positive feelings, while a lower valence value means that a song conveys more negative feelings. Clearly, the valence effects the danceability. When the musical positiveness is true (> 0.5) the median of the danceability for all playlist genres are higher. The playlist genre also effects danceability. If we compare the median of the danceabilities for the songs in a given playlist genre with a valence > 0.5, the order from highest to lowest danceability is rap, edm, pop, rock. If we do this comparison for songs with a valence < 0.5 the order remains the same.

```r
anova(lm(danceability ~ playlist_genre + musical_positiveness,data = temp))
```

```
## Analysis of Variance Table
##
## Response: danceability
##                         Df Sum Sq Mean Sq F value    Pr(>F)
## playlist_genre           3 107.16  35.719  2279.8 < 2.2e-16 ***
## musical_positiveness     1  38.08  38.076  2430.2 < 2.2e-16 ***
## Residuals            22242 348.49   0.016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For playlist genre, P value is too small, so we reject the hypothesis that the four playlist genres have equal population means.

For valence, P value is too small, so we reject the hypothesis that the four playlist genres have equal population means.

This implies that playlist genre and musical_positiveness (valence) is a significant factor that changes the danceability. Thus we reject the hypothesis that the population means among different playlist genres and valences are equal.